Searching the WWW with Mulitple Mobile Agents

D.J. Grey
University of Hull
Department of Computing, Hull, UK,
+44 (0)1482-465910
d.j.grey@dcs.hull.ac.uk

P. Dunne
University of Sunderland
School of Computing, Engineering and Technology, St. Peter's Campus, Sunderland, SR6 0DD, UK.
+44 (0)191-515-2754
peter.dunne@sunderland.ac.uk

R.I. Ferguson
University of Sunderland
School of Computing, Engineering and Technology, St. Peter's Campus, Sunderland, SR6 0DD, UK.
+44 (0)191-515-2754
ian.ferguson@sunderland.ac.uk

ABSTRACT

This position paper describes ongoing work on an agent-based approach to searching for information on the World Wide Web (WWW). The reasons why current searching techniques are inadequate and the rationale for using an agent-based approach are briefly presented. A discussion of the architectural design of an agent-based searching system is given. The system has been simulated in order to try out various agent interaction strategies. The results of the simulation have influenced the design of a prototype implementation, which is also described.

KEYWORDS

agent, mobile-agent, searching

1. INTRODUCTION

The current means of locating information on the WWW relies on the use of centralised indices known as “search engines”. These engines attempt to maintain the currency of the information they hold by a variety of means based on spiders, webcrawlers etc.[5] These indices are then queried by users to find information on a particular topic. The use of centralised indices means that these machines form a bottleneck when attempting to locate information. The size of the indices and the processing power required to service search requests, coupled with the ever increasing size of the data corpus to be indexed casts some doubt on the ability of the search engines’ technology to continue to cope [1].

At any one time, any given search engine is estimated to cover no more than 40% of the web in its database[1] [8]. To perform an exhaustive search requires the user to employ several search engines and to assume that each one has access to a different 40%

To remove the bottleneck phenomena, the index needs to be decentralised (distributed) in the same manner that the raw information is. The growth in interest in mobile agent systems [6],[11] leads the authors to suggest that one possible means of achieving this is by using mobile agents to wander the web (in a directed fashion rather than random “wandering” however) seeking the information on behalf of the user. In order to avoid pointless random wandering, some means of directing an agent towards the information it seeks is required.

The proposed scheme (termed the AgentSeek system) involves three separate types of mobile agent - “ferrets”, “publicists” and “gurus.”

2. DISTRIBUTED INDEXING

2.1 What is a Distributed Index?

In a distributed indexing scheme, each website is responsible for maintaining its own index - i.e. a list of URL’s against subject. The process of producing the index could be manual, but a more satisfactory approach would be to automate this. Thus each webserver maintains a small part of the overall index.

2.2 Why Does it Solve the Problems?

As a distributed index is not held at a central point, the search requests from the whole Internet are not concentrated in one portion of the network and thus the problem of bottlenecks can be avoided. Since the re-indexing process is local, it can be performed under the control of the site creator as often as required. This means that the currency of the index can be much better. Shifting the onus of the indexing task to the information provider should present no problem on the basis that having bothered to place the information on the web, the site creator presumably wants to ensure that it can be found. Coverage should therefore improve to the extent that website creators want it to.

2.3 How Does a Distributed Index Work?

An index is, in essence, a centralised point of reference. How can an index be distributed without encountering the very same problem it is designed to solve - where do you look for the required piece of the index?

This is where mobile, collaborative agents come in. At the simplest conceptual level the AgentSeek agents are pieces of code that traverse the Internet looking at the indexes to find the right information. They (hopefully) meet other agents looking for the same thing and share knowledge. The details of how and why this scheme works form the following sections.

3. PERFORMING DISTRIBUTED INDEXING WITH AGENTS

Before fully describing the searching algorithm, it is necessary to describe something about mobile agents as implemented in the AgentSeek system. AgentSeek is implemented on top of a mobile agent system known as AgentSpace[3].

3.1 What is an Agent?

An AgentSpace agent is a piece of mobile code [11] [6] and associated data which can move itself from one machine to another. In fact it is a mobile object (in the object oriented sense). Each agent has a unique identifier or “handle” which can be used to locate and communicate with the agent at any time.

An agent can only exist on machine that is running a piece of software known as an agentserver. Agentservers accept agents to “live” in their address space. Collectively the spaces are known as agentspace. Agents can move from one agent server to another (following either a set itinerary or learning their route as they go) forming a web of agentservers which is the mirror of the information (hypertext) web.

In the AgentSeek system an agentserver is associated with each participating webserver. In fact AgentSeek has its own built-in webserver. Associated with each website (or agentspace) is a site index - effectively a list of the pages on the site and which subject each pages is about.

3.2 Ferrets, Publicists and Gurus

Agents represent real people or organisations in cyberspace. They are able to autonomously act on behalf of their human counterparts and can negotiate with each other in order to achieve their goals.

Three types of agent are required by the AgentSeek system; “ferrets” which seek for information providers and advertise the location of information consumers, “publicists” which advertise the location of information providers and seek information consumers and “gurus” which facilitate encounters between ferrets and publicists. Ferrets and publicists have “interests” i.e. details of the subject area they are seeking or publicising. The missions of the ferrets and publicists are identical - simply to meet agents of the opposite type with the same area of interest. Whether they are seeking or publicising is irrelevant. Gurus do not represent anybody - they are the “memory” of the system - having knowledge of other agents.

Users looking for information (web searchers) send ferrets out “around the web”. People providing information (web site creators) send publicists out “around the web”

The role of the guru agents is to engender meetings between agents with similar interests, be they ferrets or publicists. They achieve this by remembering the handles and interests of agents they have met and pass them on to any agents the guru meets with similar or related interests.

When an agent arrives in an agentserver it talks to a (static) space manager agent to find out what other agents are present in that space. It then interacts with each of those other agents - finding out their interests. If it meets with an agent with similar interests, it can exchange details and amend its itinerary accordingly. The details exchanged can be addresses of web pages it has found and the addresses of sites where it is likely to meet more like minded agents or the agent handle of other agents it should attempt to contact. An agent can contact another agent directly (using the AgentSpace messaging service once it has the target agent’s handle.

The interaction (or dialogue) between agents is at the core of the system. The mobility, data carrying and processing facilities of the agents are all provided by the underlying agent authoring system (AgentSpace). It is in the interaction (or dialogue) between agents that the key functionality of the system lies.

3.3 Indexing with Agents

The means of producing the index for a website is beyond the scope of this paper as it is irrelevant to the operation of the ferrets, publicists and gurus. One possible approach is to use a fourth type of agent - an indexer - which implements some form of automated classification scheme.

The classification scheme too, is irrelevant to the operation of the agent population. The whole issue of ontology - the idea of agents having a common means of describing their interests - is the subject of much research [9]. Current work on AgentSpace instead has concentrated on the problem of getting the agents moving around and meeting.

AgentSeek (or at least its next implementation) could draw on several areas of existing technology for it classification scheme (means of describing a web page’s subject or an agent’s area of interest):

i) The current implementation uses the Dewey Decimal system. Automated Dewey classification could be built into an indexing agent.[7]

ii) The keyword approach could be adapted from existing search engines

iii) Automated understanding of text (sense tagging) could provide a framework for describing subject areas

iv) More intelligent indexing schemes could be based on describing the areas of interest in XML or some associated metadata format.

3.4 Searching with Agents

The way the system achieves its goal of locating webpages for users is best explained by considering how the agents behave when only some of the components of the population are in place and then adding other components and examining the dialogue between them

3.4.1 System with just Ferrets

The simplest (and prohibitively inefficient) case is that where ferrets move from agentspace to agentspace looking directly at each index and returning to their human master when any relevant pages have been found.

At a practical level ferrets can be launched on their way from either a command line based interface or, more comfortably, from a web page which takes its inspiration from the current search engines interfaces'.

In this scenario, the ferrets' itinerary must be preset, as they have no means of learning where to go. Theoretically they would just be given a list of all known agentservers and it is apparent that given sufficient time they would traverse every server finding every relevant page.

3.4.2 System with Ferrets and Publicists

A more complex (but still impractical) system is that with a population of both ferrets and publicists. In this scenario, ferrets interact with the publicists. Publicists are launched knowing about a set a pages on a given area of interest. Again preset itineraries are required. By relying simply on chance encounters and allowing sufficient time, ferrets and publicists with similar interests will meet and be able to exchange information about the location of the relevant web pages.

Two different types of publicist exist; "selfish" and "altruistic". Selfish publicists when interacting will pass on only details of the website they were sent out to advertise, "altruistic" publicists will pass on details of all the sites they know of about their particular topic.

3.4.3 System with Ferrets, Publicists and Gurus

It can be seen that the problem of reducing the time taken to perform a search to practical proportions is one of population density - How many agents are required to be navigating around how many sites in order to increase the probability of sufficient like minded agents meeting (thus allowing sharing/learning of information) in a reasonable timescale?

3.4.4 Interaction Strategies

Several different agent itinerary modification strategies (interaction strategies) are possible:

i) ferret-ferret interactions - When two ferrets seeking the same topic meet, they can exchange details of sites they have found matching the topic.

ii) ferrets - publicists interact - When a ferret and publicist with an interest in the same topic meet, they will exchange details of sites they have found matching the topic.

iii) publicists - publicist interactions - When two publicists advertising the same topic meet, they can exchange details of sites they are aware of matching the topic.

iv) any combination of the above behaviours

A successful interaction strategy is likely to use indirection to engender "small-world" effects (described by Strogatz [12]) and utilise them to minimise the number of interactions required to locate information on a given topic.

4. PROGRESS IN IMPLEMENTING THE SOLUTION

At the time of writing, AgentSeek exists in prototype form only. Its development has been informed by results of using a simulation of the algorithm. This section describes both simulation and prototype.

4.1 The Simulation

That the system outlined in the previous section has the emergent property that it allows like-minded ferrets and publicists to meet given infinite time to wander around is self evident. Several questions then arise such as:

- How many ferrets, publicists and gurus are required in a population of agents to improve to a useful level the chances of appropriate ferrets and publicists meeting and exchanging information in a useful timescale?
- What interaction strategy should be adopted by the agents to maximise their searching capabilities.

In order to gain some understanding of the answers to these questions the AgentSeek system has been simulated by a simple (C++) program that;

- represents the servers as a collection of nodes
- has each node flagged as possessing a topic or not
- has a number of ferret and publicist objects present at each node

The model is initialised by choosing:

- the number of nodes
- the percentage of those nodes with a topic present
- the number of ferrets in the system, each of which is given a random starting point and initial itinerary
- the number of publicists each topic node launches. Each publicist is given an initial itinerary at random.

The model then evolves in discrete steps. At each step:

- ferrets at a node
  - examine the node to see if the topic is present or not
  - can exchange information with other ferrets present at the node
  - can obtain the home site of any publicist present at the node
- ferrets and publicists are moved to the next node on their itineraries
  - a ferret stops travelling (and returns home) when it reaches the end of it's itinerary
  - a publicist starts again at the beginning of it's itinerary when it reaches the end

By choosing appropriate initial values for the model we can observe the average success rate of the ferrets and their average path lengths.

4.2 Prototype

The prototype of the AgentSeek system seeks to model the activity of ferret, publicist and guru agents operating on the WWW. The prototype is implemented using the same technology (the AgentSpace mobile agent system) as the full-scale AgentSeek system. The prototype however differs from the final full-scale system in several important aspects:

i) - The agents are restricted to a limited number of workstations (10 - 30).

ii) - No real webservers or webpages are used, instead the action of a webserver is simulated by the use of a static agent which can tell other agents how many pages it has on a given topic. Since no real webpages are involved, no automated categorisation of subject takes place.

iii) Due to the use of simulated webservers, the distribution of pages and subjects between those servers has been simulated as well. In simulating this, it was assumed that pages on a given topic are clustered on certain servers rather than being randomly distributed.

iv) The agents do not carry any real information about. They simply know that they are looking for or advertising a particular Dewey classmark. They record the number of servers visited, agents met and whether or not these matched the topic they sought.

v) The matching take place when a publicist arrives on an agentserver and promptly informs every other agent there present about the topic it seeks.

By using actual agents, the random meetings between agents are simulated in a timescale not untypical of the real scenario being simulated.

4.3 Experimental results

A series of experiments was performed with the aims of;

i) testing the validity of the simulation

ii) gaining an understanding of how agent population size and interaction strategy affect the efficiency of the system.

Details of the experiments and their results have been given elsewhere [2], but the findings can be summarised thus:

i) Results obtained from the prototype and the simulation running under identical conditions are comparable. This implies that the numbers from the simulation can be trusted.

ii) The approach does work. The agents will locate the resources they seek.

iii) The system will achieve useful results in a usable timescale.

iv) The various interaction strategies fall into 2 categories, ones which obtained an efficiency of ~10% (each ferret discovered 10% of the relevant pages available) and a second which obtained 80-90%. The significance of this result to the development of AgentSeek is explained in the following section.

5. IMPROVING SYSTEM PERFORMANCE

5.1 Improving altruistic behaviour

To be a viable alternative to current search technology, the number of agents and time taken to perform high quality searches must be minimised. To do this, the path that an agent must traverse to find the information it seeks (the number of steps in agentspace that an agent is from its desired target) must be reduced. Result iv) in the previous section showed that the most effective interaction strategies are the ones where agents swap all or most of their information, i.e. altruistic rather than selfish behaviour.

The thrust of current work is to further develop the prototype system by implementing altruistic behaviour. One practical problem behind this is that anyone who has sent out a publicist, might be (justifiably) annoyed if that publicist had a part in publicising a rival site on the same topic. For this reason, altruistic publicists will probably not be implemented in the final system. One possible solution is to factor the altruistic behaviour out of the ferrets and publicist completely and let it reside solely in the guru agents. The required relative populations size to make this strategy effective is the subject of a current simulation study.

Other means of increasing the probability of agents with common interests meeting also revolve around guru behaviour. Possible strategies include:

i) Introducing a level of indirection into the system in the form of gurus which can remember other gurus

ii) Using specialist gurus, i.e. ones that remember only agents with certain interests? - The hierarchical nature of the dewey decimal coding system used in the prototype leads to the possibility of organising the gurus in some hierarchy whereby if a guru doesn’t know about a particular area of interest it can pass the query up to a higher level guru.

iii) Using mobile gurus that, spread knowledge as they go.

5.2 Replication, Breeding and Population Dynamics

An as yet uninvestigated possibility is to allow agents to spawn other agents. Agent’s itineraries tend to grow as an agent learns the whereabouts of more relevant sites via negotiation with other agents. Once an agent has a sufficiently large itinerary (list of sites to visit) it would be possible for an agent to clone itself with each resulting agent handling half of the itinerary.

One suggestion is to allow agents that have been particularly successful to somehow breed, presumably by merging their knowledge and then replicating, whilst terminating others which have itineraries that have failed to allow them to pass on their knowledge.

Such behaviours would endow the system with if not intelligence at least some form of adaptivity which may be beneficial to performance. The performance of such a agent system is as yet unknown (both in terms of population dynamics and effectiveness) but should be amenable to study by simulation techniques.

6. CONCLUSION

The “ferret, publicist, guru” paradigm for searching the web is likely to work. The variables in the scenario are manifold and some kind of “what if” analysis needs to be performed on several more competing interaction strategies before an implementation of the full system can take place

The exchange of information between agents is a “mixing phenomenon.” The more information that is passed around the system, the more likely it is that a given agent can satisfy its goals. More simulation work is required to develop more effective interaction strategies. One possible strategy that will be investigated is to devolve all “altruism” in the system to specialist guru agents.

The practise of simulating an agent based system before fully implementing it has proved extremely valuable in gaining insight into the system’s behaviour.

Other future work will include implementing the full system and generalising the design into a pattern for any scenario where two agents are attempting to meet - an “agent dating agency” pattern.

7. OBTAINING THE SYSTEM

AgentSeek [4] and the underlying toolkit AgentSpace [3] are both available for downloading from their respective websites.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the efforts of Mr. A. Wyvill and Mr. J. Cartledge, who keep our network running despite our experiments

REFERENCES

[1] - Bharat, K & Broder, A, *A technique for measuring the relative size and overlap of public Web search engines*, Computer Networks and ISDN Systems, Vol 30, No. 1-7, pp 379-388, 1998

[2] - Grey, D.J., Dunne, P. & Ferguson, R.I., AgentSeek: a means of efficiently location resources on the World Wide Web using mobile, collaborative agents. Awaiting publication in Proc. ASIM/SCS Workshop 2000, Agent-based Simulation, University of Passau, Germany, May 2000.

[3] - Grey,D.J. & Ferguson, R.I., “AgentSpace”, Online, http://www.cet.sunderland.ac.uk/~agent/agentspace/welcome.html download Feb, 1999

[4] - Grey,D.J. & Ferguson, R.I., “AgentSeek”, Online, http://www.cet.sunderland.ac.uk/~agent/agentseek/welcome.html download Feb, 1999

[5] - Hock, R., *Web search engines: Features and commands: Search Engine Section*, Online, 1999, Vol.23, No.3, pp.24-28

[6] - Huhns, M.N. & Singh, M.O., *Agents on the Web*, IEEE Internet Computing, May-June, 1997

[7] - Jenkins, C. Jackson, M., Burden, P. & Wallis, J., Automatic classification of web resources using Java and Dewey decimal classification, Computer Networks and ISDN Systems, Vol 30, p646-648,1998

[8] - Lawrence, S. & Giles, L., Accessibility of Information on the Web, Nature. Vol 400, No. 6740, p107-109, 1999

[9] - Luke, S. Spector, L. Rager, D. & Hendler, J., Ontology-based Web Agents, In Proc. of the First International Conference on Autonomous Agents, 1997

[10] - Novak, L & Gibbons, *Hybrid Graph Theory and Network Analysis*, Cambridge University Press November 1999

[11] - Pham,V.A & Karmouch, A., *Mobile Software Agents: An Overview*, IEEE Communications Magazine, pp26-37, July, 1998

[12] - Strogatz, S,H., *It’s a small world,* Nature, vol. 393, no. 6684, p409-410